REMARKS

In view of the foregoing amendments and following remarks, Applicant respectfully requests favorable reconsideration of this Application.

No claim amendments are proffered herein.

In the present Office Action, the Office stands by the rejections made in the previous Office Action, in which all of claims 1-29 were rejected as anticipated under 35 U.S.C. §102(b) by Lucus.

The Present Invention

The present invention is a method and apparatus for automatically opening files of particular types on a computer using attributes such as window size and window position dictated by how the user previously positioned and sized windows when viewing files of the same type.

It further includes the concept of, when a user opens a certain file of a first type (the first file), automatically opening a second file that has some file name attribute relative to the file name of the first file. For instance, whenever a file having a particular given name with a first file type extension, e.g., johnsmith.doc, is opened, the computer will automatically open a second file having the same file name but a file type extension of a second type, e.g., johnsmith.pdf. The invention is particularly useful for users who repeatedly open one or more files of certain types that they would like to be sized and positioned in the same place every time and/or repeatedly need to open two related

files and view them simultaneously, such as might be necessary for repetitive data entry tasks.

In short, the software of the present invention remembers at least one display attribute of a file being used by a user, for instance, the position and size of the window in which the file is displayed. Then, when the user opens another file of the same file type, it will automatically open in a window in the same position and of the same size as the previous file of the same type. The invention can be applied to several different file types so that a user can open multiple files that he/she may need to view simultaneously and they will always open up in the position and size windows that the user desires.

The Lucus Reference

The Lucus reference pertains to the display of computer files and documents on a computer display. Furthermore, it pertains to the permanent storage of "ephemeral" attributes of a document, including display attributes (such as its coordinates on the display screen) for re-use when the document is displayed a next time. However, contrary to the present invention, Lucus has nothing to do with applying those display attributes to other files of the same type (which is a core concept of the present invention).

More specifically, Lucus discloses a computer-controlled information management system in which documents in the system have permanent attributes and ephemeral attributes, each attribute having a name and a value. Attributes that are normally permanently stored are called permanent attributes. Attributes that typically

are created only when the document is being displayed, such as information regarding the position on the display unit, are called ephemeral attributes. In accordance with the disclosure of Lucus, ephemeral attributes may be converted into permanent attributes and stored with the documents after the user is done referencing or modifying them.

In accordance with another aspect of Lucus that the Office deemed significant, Lucus employs a feature referred to as "strands". In accordance with this feature, multiple documents are collected together as a strand and can be treated as a unit. However, again, a strand is a predefined set of documents. There is no discussion whatsoever in Lucus of applying the display attributes of one document to another document.

Traversal of Prior Art Rejections

In the present Office Action, the Office has repeated the previous rejections and additionally has replied to Applicant's arguments presented in response to the previous Office Action. The rejections and Applicant's previous responses will not be repeated here. Reference may be made to Applicant's previous response for background information. The Office's reply is threefold. Applicant will herein respond to the Office's three points in order.

1. Lucus does not teach the relationships between documents based on document types

In reply to Applicant's previous arguments, the Office asserted:

> Lucus clearly teaches a Unique Identifier, or UID, is a string of alphanumerics that uniquely identifies a document. A UID is necessary and sufficient to refer to a specific document (col. 4 lines 7-23; the attributes define

> the display characteristics of an associated document, such as position and size (col. 6 lines 61-66), and by using the UID to define the appearance and location of the documents (col. 10 lines 23-33); therefore, the documents can be retrieved and displayed with the same setup (or layout) of the given set of parameters is matched its type or ID.

It appears that the Office is misinterpreting Applicant's point here. Applicant's point here was merely the general point that Lucus does not disclose taking display attributes that are set in connection with a first document and applying them to another document based on that other document being of the same file type as the one document. The Office appears to be arguing that Lucus discloses relating appearance attributes with a document. Applicant does not dispute this. In Lucus, those attributes are used with the single document with which they are associated. Applicant's point is that Lucus does not teach taking those attributes that are assigned to one document and applying them to another document.

As noted above, this is Applicant's general point. The second and third points addressed by the Office are the details that support Applicant's first point.

2. Lucus fails to teach applying attributes of one file to any other file

In addressing this point in the final Office Action, the Office asserted "Lucus clearly discloses matching types of documents if they are in same characteristics based on UIDs, input parameters, and attributes (e.g., col. 2 lines 18-32)".

Column 2 lines, 18-32 of Lucus state:

> Each time a user requests a document, the client sends a search request to one or more repositories. The repositories respond with one or more messages containing the unique identifier(s) of documents that match the description in the search request. The client then requests permanent attributes of the documents corresponding to the received UIDs, and the repositories respond by sending

> requested permanent attributes of the requested document to the client. The client then determines whether any of the received permanent attributes for that document are actually ephemeral attributes defining the previous visual display of the document, stored as permanent attributes in the repository. The client converts such permanent attributes into ephemeral attributes and uses their values to create a display of the document on a display device.

Again, the Office seems to have misinterpreted Applicant's point in this argument, the nature of the present invention, and/or what is disclosed in this paragraph of Lucus. As a preliminary point, Applicant is unsure what the Office means by the term "matching", but will proceed on the assumption that it is referring to the process of retrieving documents based on their meeting, or "matching", certain search criteria.

The fact that the Office has responded to Applicant's argument that "Lucus fails to teach applying attributes of one file to any other file" by asserting that "Lucus clearly discloses matching types of documents if they are in same characteristics based on UIDs, input parameters, and attributes" suggests a fundamental misunderstanding as to what is being claimed because applying attributes of one file to another file does not have anything to do with "matching" files by their attributes (or UIDs, for that matter). In fact, it has nothing to do with "matching" files at all.

Applicant is not disputing whether Lucus teaches matching documents by file type. Certainly, this is old technology. In fact, any file search software can match files by type. All one would have to do is run a search for files containing a particular file extension.

On the other hand, this paragraph does not teach matching files by attributes or UIDs, as the Office seems to have asserted. It merely teaches that the documents that matched the search criteria are listed by their UIDs, not that the UIDs are the search parameters. Furthermore, this paragraph does not teach using the display attributes to match (i.e., retrieve) files as the Office seems to have asserted. Furthermore, the Office's assertion is troubling, not just because it is not true, but because it is irrelevant. Specifically, whether Lucus teaches retrieving files by their display attributes is irrelevant. The present invention does not have anything to do with retrieving (or matching) files by their display attributes. The present invention concerns displaying a file using the attributes assigned to a different file of the same type). This is nowhere found in Lucus.

The above-quoted paragraph of Lucus discusses two essentially unrelated aspects of Lucus that occur sequentially, and the Office seems to be combining them simultaneously using hindsight reconstruction in a way that is totally different than what is actually disclosed in the paragraph. Specifically, in this paragraph, Lucus describes retrieving a plurality of documents that have something in common with each other (as defined by the query). For instance, they may all be .jpg files. The system returns as a result a list of the UIDS of all of the files that meet the criteria of the search. Then, the client asks for and receives the permanent attributes of the files/UIDs in the list. The client then displays each file using the retrieved attributes for that file. It does not retrieve the files by these attributes (and even if it did, it would be irrelevant to the

present invention). It does not apply the attributes of a file to any file in the list but the file to which those attributes are assigned.

Nothing in this paragraph talks about using the attributes of one file in the list to display a different file in the list. Each file in the list is displayed by using its own attributes. This is the opposite of taking the attributes of one file and using them on other files.

3. Lucus fails to teach displaying another file using the same value of the first file

In addressing this point in the final Office Action, the Office asserted "Lucus' invention clearly teaches that every document can be defined by size variables as input parameters which use to determine the size and location the document on the display device (col. 7 lines 26-35); therefore, the other (or second) document will use the same value of the first one when displaying on the screen".

Column 7, lines 26-35 of Lucus are reproduced below for reference:

> For example, every document has a position in world space defined along the x, y, and z axis, and every document has a width and a height. When an image of the document is drawn on the display device, the perspective function takes those world space coordinates and size variables as input parameters, and determines the actual size and location on the display device, in "screen space coordinates", where the document is actually going to be drawn. The perspective function is instantiated by the workspace viewer process.

This paragraph discloses nothing more than that each document is displayed in some defined position on the screen and using the display attributes assigned to that document. It does not discuss how the document positions are defined. Therefore, this is irrelevant (and it does not appear that the Office is even asserting that this portion is

relevant). It does discuss how the size of the document is determined (this appears to be the portion upon which the Office is relying), but it merely discloses that the "ephemeral" display attributes of each document are used to display that document. Thus, Lucus does not disclose applying the "ephemeral" display attributes of one document to another document based on the document type. In fact, Lucus does not disclose applying display attributes of one document to another document, period.

Distinguishing Claim Language

Accordingly, with reference to independent claim 1, Lucus does not teach either of steps (3) and (4), which recite "when another file of the type of said first file is opened by an operator for display, accessing said stored data indicating said value of said at least one attribute" and "displaying said another file of the type of said first file using the same value of said at least one attribute as said first file".

Dependent claim 12 further distinguishes over Lucus by adding the feature of opening a second file of a certain file type whenever a first file of another file type is opened, the second file having the same defined relationship with the first file as the two files previously displayed simultaneously.

Lucus clearly does not disclose what is claimed in claim 12. While Lucus does appear to disclose opening a group of files with one command, it opens the second and subsequent files simply because they are in a predefined workgroup and not for any reason relevant to file types.

Lucus, therefore, does not teach the limitations of steps (5) and (6) of claim 12. Particularly, there is nothing in the workgroup concept of Lucus that teaches "storing

data associated with said type of said first file indicating at least a type of said second file relative to said first file". Rather, in Lucus, there is simply a list of specific files in a workgroup.

Furthermore, Lucus does not teach "when another file of the type of said first file is opened for display, automatically opening another file of the same type as said second file pending the same relationship to said another file of said first type as said second file had to said first file". As previously mentioned, Lucus discloses no concept of applying attributes from one file to another file, let alone doing so based on file type.

Independent claim 21 recites subject matter similar to claim 1 and, thus, distinguishes over Lucus for essentially the same reasons as claim 1. Particularly, claim 21 recites "using the same value to display said another file", that value being previously defined as "a value of at least one display attribute of a first displayable file on a computer".

Dependent claim 25 recites subject matter similar to dependent claim 12 discussed above and, thus, even further distinguishes over the prior art for the same reasons given above in connection with claim 12.

All other claims depend from one of claims 1 and 21. Accordingly, they distinguish over the prior art for at least the same reasons.

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice

of Allowance at the earliest possible date. The Examiner is invited to contact Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: 9.27.04

Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: 215-923-4466